



ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 299 (2003) 585–602

Theoretical
Computer Science

www.elsevier.com/locate/tcs

An asymptotic study for path reversal

Alain Giorgetti

*Laboratoire d'Informatique de l'Université de Franche-Comté, 16 route de Gray,
25030 Besançon Cedex, France*

Received 12 November 2001; received in revised form 22 May 2002; accepted 4 June 2002

Communicated by M. Jerrum

Abstract

A path reversal is performed in a rooted tree when a node becomes the root of all the nodes along the path from it to the former root. This algorithm on trees is presented as a transition system specified by induction over a convenient view of the tree structure. When each tree node is assigned a fixed weight representing its relative probability to move to the root, the transition system defines a finite Markov chain. This paper presents some of its asymptotic properties. A closed formula for the stationary distribution and a tight upper bound for the average computational complexity of path reversal are also given as new results.

© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Rooted tree; Path reversal; Markov chain; Average-case complexity; Stationary distribution

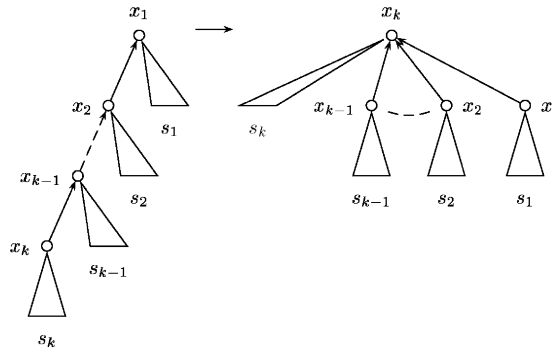
1. Introduction

Coming from graph theory, with the addition of one distinguished vertex called the root, trees are common in computer science. As data structures, they often lead to efficient algorithms. For instance, they organize data in directories within operating systems and they structure classes in object oriented design. Algorithms handling trees can be divided in two classes. Those from the first add and remove nodes and those from the second only change the tree structure for a fixed set of nodes. This paper studies a basic algorithm of the second kind that appears independently in many areas of computer science.

E-mail address: giorgett@lifc.univ-fcomte.fr (A. Giorgetti).

0304-3975/03/\$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.

PII: S0304-3975(02)00537-6

Fig. 1. Path reversal at node x_k .

Path reversal was first presented as a kind of path compression providing an efficient solution to maintain disjoint sets under union [3,11]. Path reversal also inspired an efficient distributed mutual exclusion algorithm [1,10] which can perform a mutually exclusive access to a shared resource. In its non-concurrent version, the algorithm appears to be a path reversal over a logical tree structure between requirers. In this applicative framework, the average number of messages per request is significant. This average cost has been proven [5,6] to be harmonic in the case of equiprobably requiring devices and has been conjectured to be lower in the more general non-equiprobabilistic case [9]. This conjecture is solved in this paper as a consequence of a general study of the stationary properties of path reversal.

The path reversal initiated by node x_k in a tree t modifies the tree by making each node on the path from the parent of x_k to the root x_1 become a child of x_k . This operation is illustrated by Fig. 1, where the triangles s_1, \dots, s_k denote subtrees with respective root x_1, \dots, x_k .

The cost of a single operation can be defined by (or closely related to) the number of oriented edges that it moves. When a sequence of m consecutive reversals is performed on an arbitrary initial tree with n nodes, a worst-case complexity can be defined as the maximal cost for such sequences. This transient analysis has been carried out in [3].

The main focus here is on the asymptotic complexity in the average case, that reflects the algorithm behavior over the long-run. The idea of an average behavior makes sense when each node is assigned an arbitrary probability to induce a reversal. The distribution of node probabilities turns the path reversal into a Markov chain. In the equiprobabilistic case (when nodes do not need to be distinguished since each one induces reversals with the same probability), it has been proven by different means [5,6] that the algorithm requires only H_{n-1} edge moves on average (order of $\log n$), in the mutual exclusion context where the tree size n is also the number of competitive requirers.

This paper studies the general stochastic case where the equiprobability hypothesis is relaxed, with the aim of learning more about the limiting properties of a large number of consecutive reversals. Since probabilities to reverse are only related to nodes (not to their location in the trees), rooted trees where all the nodes are distinguished by

a different number, called their *label*, are needed. These labeled trees are defined in Section 2 as algebraic structures that provide a compact description of path reversal as a transition system in Section 3. Probabilities are introduced in Section 4 for a stochastic version of path reversal. Asymptotic properties are stated in Section 5. The stationary state distribution is one of the new results derived in this paper. In Section 6.1 the average-case asymptotic complexity is defined and an inductive definition is given for its computation for any tree size n . This definition is used in Section 6.4 to get a tight bound for this complexity. Finally, Section 6.5 interprets these results in the original context of mutual exclusion.

2. Rooted trees

This section defines the data structure handled by the operation of path reversal. These *oriented trees* [4, p. 306] are called here *rooted trees*, since the rooting action that distinguishes one vertex in a graph-theoretic tree uniquely determines an orientation of all the tree edges.

A *rooted tree* is inductively defined as a non-empty set of nodes with one distinguished node called the *root* and a possibly empty set of *direct subtrees* that are rooted trees partitioning the remaining nodes.

From this definition, two rooted trees that only differ by the ordering of their subtrees are considered to be identical. There are two reasons for the choice of such *unordered* trees. The first is that unordered trees directly represent the structures of the algorithms that use path reversal. In the union-find problem [8] they represent disjoint sets. In the mutual exclusion context [10] they represent a logical structure that has no spatial meaning. The second reason is an aesthetic one: this choice leads to more attractive and simpler proofs than the well-known inductive structure of planar (ordered) trees.

A *node* in a rooted tree t is either the root of t or a node of a direct subtree of t . A *subtree* in an abstract rooted tree t is either the tree t itself or a subtree in a direct subtree of t .

2.1. Notations

Let RT denote the set of rooted trees. For every tree t in RT , let $r(t)$, $c(t)$, $n(t)$ and $s(t)$, respectively, denote the root, the direct subtree set, the node set and the subtree set of t . $t = [r, c]$ can be written to say that t is the tree rooted at r and with the set c of direct subtrees. In short, the relation $u \in s(t)$ is denoted $u \subseteq t$. For every s in $c(t)$, let $t - s$ denote $[r(t), c(t) - \{s\}]$ and, for every s in RT , let s/t denote $[r(t), c(t) \cup \{s\}]$.

Using these notations, the following is always obtained:

$$n(t) = \{r(t)\} \cup \bigcup_{u \in c(t)} n(u), \quad s(t) = \{t\} \cup \bigcup_{u \in c(t)} s(u) \quad (1)$$

with $\bigcup_{u \in \emptyset} = \emptyset$,

$$r(s/t) = r(t), \quad c(s/t) = \{s\} \cup c(t) \quad (2)$$

with $c([r, \emptyset]) = \emptyset$ and

$$n(s/t) = n(s) \cup n(t), \quad s(s/t) = \{s/t\} \cup s(s) \cup s(t) - \{t\} \quad (3)$$

with $n([r, \emptyset]) = \{r\}$ and $s([r, \emptyset]) = \{[r, \emptyset]\}$.

2.2. Node labeling

In all that follows, a *label set* L is a non-empty finite subset of $\mathbb{N} - \{0\}$ and $|L|$ denotes the *size* of L , defined as its number of elements.

Let n be a positive integer and L be a label set with $|L| = n$ integers.

Definition 1. An L -labeled rooted tree is a rooted tree t with $|L|$ nodes and a bijection between the node set $n(t)$ of t and the label set L .

The set of L -labeled rooted trees is denoted $A(L)$. The set of all the $\{1, \dots, n\}$ -labeled rooted trees is denoted A_n . Let A denote the disjoint union of $A(L)$ for L ranging over the non-empty finite subsets of $\mathbb{N} - \{0\}$:

$$A = \bigcup_{L \subset \mathbb{N} - \{0\}, L \text{ finite}, L \neq \emptyset} A(L). \quad (4)$$

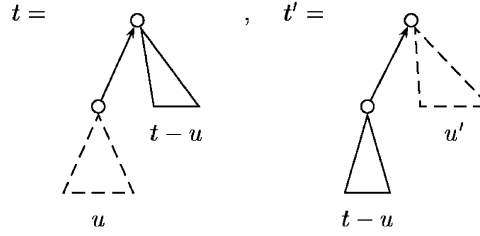
For notational convenience, each node is identified with its node label and the labeled rooted trees of size 1 are identified with their root label. Thus, for every integer $i \in L$ and every tree $t \in A(L)$, $i \in n(t)$ or $i \in t$ can be written. Node sets are also identified with their corresponding label sets. Thus, for every labeled tree t in A , $t \in A(n(t))$ can be written.

When S and T are disjoint label sets, two labeled rooted trees $s \in A(S)$ and $t \in A(T)$ can be composed to form the labeled rooted tree $s/t \in A(S \cup T)$. This obvious composition rule provides an inductive definition of $A(L)$ using the subsets of L , that will be made explicit later in this paper.

3. The path reversal algorithm

The path reversal algorithm (PRA) can be defined as an operation over labeled rooted trees. Since labels are statically linked to nodes, it is sufficient to describe this algorithm over rooted trees: each label simply follows its moving node. Furthermore, it serves no purpose to indicate which node induces the reversal, since it becomes the root in the resulting tree.

In this section the path reversal algorithm is described as a transition system whose states are rooted trees. A *transition system* $S = (Q, R)$ consists of a set Q of states and of a binary relation $R \subseteq Q \times Q$, called the *transition relation*. Transition systems can represent indeterministic algorithms, $(q, q') \in R$ meaning that one goes from state q to state q' in one algorithmic step.

Fig. 2. Illustration of $(t, t') \in \theta$ with $(u, u') \in \theta$.

For the PRA, the transition relation translates the indeterministic choice of the next reversing node. This relation accepts a simple inductive definition, using the notations introduced in Section 2.

Definition 2. The *path reversal transition system* is the pair (A, θ) where the transition relation $\theta \subseteq A \times A$ is inductively defined by

$$(t, t') \in \theta \Leftrightarrow (t = t') \vee (\exists u \in c(t). (u, t' - (t - u)) \in \theta). \quad (5)$$

The basic case of trees with a single node is naturally included in (5), corresponding to $c(t) = \emptyset$. Fig. 2 illustrates this definition of relation θ with $u' = t' - (t - u)$. At this convenient level of abstraction, tree sizes, node labels or reversing node label are not required. The reader can easily check that this relation really does correspond to the path reversal algorithm as defined in [8] and illustrated by Fig. 1.

The following lemma may also be used as a definition for relation θ .

Lemma 3. The transition relation θ in the path reversal transition system (A, θ) is characterized by

$$(t, t') \in \theta \Leftrightarrow (t = t') \vee (\exists w \in c(t'). \exists u \in A. (u, t' - w) \in \theta \wedge t = u/w). \quad (6)$$

This lemma is a simple rewriting of the definition and can be used as a definition for the inverse relation θ^{-1} defined by

$$(t, t') \in \theta \Leftrightarrow (t', t) \in \theta^{-1}$$

and representing one algorithmic step backward from state t' to state t .

4. Algorithm stochastic form

The path reversal algorithm can be seen as a stochastic process when a probability to become the root through one algorithmic step is assigned to each tree node. This *transition probability* can be defined in the following way.

4.1. Transition probability

Let $p = (p_i)_{i \geq 1}$ be a discrete finite measure called the *node measure*. All the p_i are zero, except a finite number, whose indices form the *support* of p .

The *transition probability* is the fixed probability

$$\frac{p_i}{\sum_{i \in t} p_i}$$

assigned to each tree node i in a tree t to become the root through a single application of the path reversal algorithm on t .

The total measure $p(L)$ of a set label L is defined as

$$p(L) = \sum_{i \in L} p_i$$

and the total weight $p(t)$ of a tree t in $A(L)$ is defined by $p(t) = p(L)$.

It is not assumed that the total weight $p(t)$ is 1 in order to obtain inductive proofs (over subtrees of t). Many results that follow are stated by induction over label sets and a property stated only for L such that $p(L) = 1$ could not be inductively derived from the same property assumed for some strict subsets S of L , since $p(S) = 1$ does not generally hold.

For computational reasons, p_i will often be treated as symbols, on which the arithmetical operations $(+, -, \times, /)$ hold, i.e. computations are performed in the field $Q(p)$ of rational functions with indeterminates in p and coefficients in the set Q of rational numbers.

4.2. Markov chain

The stochastic behavior of the algorithm is completely defined by the current state (a given tree) and by the probability distribution p for a transition to the next state (a resulting tree). This defines a discrete Markov process with the number of steps from initial state as time clock. This paper limits the study to the case of fixed transition probabilities over time. Then, this Markov process becomes a *Markov chain* which is finite for each fixed number n of nodes. The first properties of this Markov chain are given in the following subsections. The main results are then obtained by a computational analysis of its asymptotic behavior and are grouped in Section 5.

4.2.1. Transition matrix

For every node transition measure p the *transition matrix* M_p of the stochastic path reversal algorithm is the restriction to non zero rows and columns of the following infinite matrix defined for every t and t' in A by

$$(M)_{t,t'} = \frac{Pr(t')}{p(t)} \Leftrightarrow (t, t') \in \theta. \quad (7)$$

4.2.2. Communicating classes of states

The strong connectivity of the state graph defined by the transition system (A, θ) was informally proved in [6]. In terms of Markov theory (see for instance [7] for a general overview), this means that the stochastic path reversal algorithm has a single recurrent communicating class, and is therefore said to be *irreducible*.

Moreover, the stochastic PRA is *aperiodic*, meaning that any state t verifies $(M_p^k)_{t,t} > 0$ for every sufficiently large k . The diagonal element $(M_p^k)_{t,t}$ of the k th power of the matrix M_p is the transition probability from t to t in k steps and is obviously positive, since $(M_p)_{t,t} > 0$.

5. Asymptotic study

As a finite irreducible and aperiodic Markov chain, the stochastic PRA converges to an equilibrium. This means that the empirical frequencies of its states (which are indeed trees) tend to a value independent of time and of the initial state, when the number of reversals tends to ∞ , forming the so-called *steady state* or *stationary* distribution.

In this section, the asymptotic behavior of the iterated PRA is investigated. Using the general result that the stationary distribution is also the unique invariant distribution (for a proof, see [7, p. 41]), an exact closed formula can be stated and demonstrated for the asymptotic probability $\pi(t)$ for the system to be in the state defined by the tree t .

The computations proceed in two steps. The first states an invariant measure for the PRA in Section 5.1. The second derives from it and gives the invariant distribution by normalization in Section 5.2. The resulting stationary distribution π gives a complete description of the algorithm behavior over the long-run. As an illustration, this vector is used in Section 6 in a quantitative analysis of an average computational cost for the stochastic PRA.

5.1. Invariant measure

For every transition measure p there is an inductive formula to compute an invariant measure λ_p for the PRA. Its support is obviously $A(L)$, where L is the support of p and λ_p can therefore be extended to the infinite set A with zero value for every tree that is not in $A(L)$.

This holds for all the functions introduced later: they all depend on a measure p that will no longer appear as an explicit index. These functions are defined on A but it is always implicit that their support is included in $A(\text{support}(p))$. As a special case, $\lambda_p(u) = 0$ is implicitly true when $p(u) = 0$ and the following definition just defines the non-zero values of $\lambda_p = \lambda$.

Definition 4. Let λ denote the function inductively defined over the set A of labeled rooted trees by

$$\forall u \in A, \quad \lambda(u) = \frac{1}{p(u)} \prod_{s \in c(u)} \lambda(s). \quad (8)$$

This inductive definition includes its base. For every $i \in \mathbb{N} - \{0\}$ considered as a labeled rooted tree with a single node, $c(i) = \emptyset$ and therefore

$$\lambda(i) = \frac{1}{p_i}. \quad (9)$$

Let S and U be two disjoint label sets. The relation

$$\forall s \in A(S), \quad \forall u \in A(U), \quad \lambda(s/u) = \frac{p(u)}{p(s) + p(u)} \lambda(s) \lambda(u) \quad (10)$$

that uses the notations and the composition rule of Section 2 can easily be derived from definition (8) of λ . This relation will be used many times in the forthcoming computations.

Theorem 5. *For every label set L and every transition measure p over L , the restriction λ_p of λ to $A(L)$ is an invariant measure for the stochastic path reversal algorithm over L -labeled rooted trees.*

Proof. It must be demonstrated that the linear algebra relation $\lambda_p M_p = \lambda_p$, where λ_p is seen as a row vector over $A(L)$ and M_p is the Markovian transition matrix defined by (7). This relation

$$\forall v \in A(L), \quad p(v) \lambda(v) = p_{r(v)} \sum_{(t,v) \in \theta} \lambda(t) \quad (11)$$

is proven by induction over label sets ordered by inclusion.

For the basic case where $L = \{i\}$ for some $i \in \mathbb{N} - \{0\}$, $A(\{i\})$ is reduced to the unique rooted tree with a single node labeled by i and (11) is obviously true.

Now, for every label set L with two or more labels, using Lemma 3 with $t = u/w$ the right-hand side of (11) can be rewritten as

$$p_{r(v)} \lambda(v) + \sum_{w \in c(v)} \sum_{(u,v-w) \in \theta} p_{r(v)} \lambda(u/w). \quad (12)$$

Then, applying (10) twice, first from left to right to decompose $\lambda(u/w)$ and second from right to left to recover $\lambda(w)$, the following is obtained:

$$p_{r(v)} \lambda(v) + \sum_{w \in c(v)} p(w) \frac{\lambda(w)}{p(v)} p_{r(v)} \sum_{(u,v-w) \in \theta} \lambda(u), \quad (13)$$

since $t = u/w$ and $(t, v) \in \theta$ implies $p(v) = p(t) = p(u/w) = p(u) + p(w)$. Now, using the induction hypothesis of (11) applied to label set $n(v - w)$, strictly included in $n(v)$, gives

$$p_{r(v)} \lambda(v) + \sum_{w \in c(v)} p(w) \frac{\lambda(w)}{p(v)} p(v - w) \lambda(v - w) \quad (14)$$

and (10) applied to recompose $\lambda(v)$ finally leads to

$$p_{r(v)}\lambda(v) + \left(\sum_{w \in c(v)} p(w) \right) \lambda(v), \quad (15)$$

which equals $p(v)\lambda(v)$ by the definition of $p(v)$, $r(v)$ and $c(v)$. \square

5.2. Stationary state distribution

The PRA algorithm converges to an equilibrium where each state has a fixed probability of visit. This *stationary (state) distribution* can simply be obtained from any invariant measure by normalization, as is done in this section.

Theorem 6. *For every label set L and for every transition distribution p over L , the stationary state distribution π_p of the probabilistic path reversal algorithm over $A(L)$ is the restriction to $A(L)$ of the function π from A to $Q(p)$ uniquely defined for every $u \in A$ with $n+1$ nodes ($n \geq 0$) by one of the three following equivalent equations:*

$$\pi(u) = \frac{\lambda(u)}{n!} \prod_{i \in u} p_i, \quad (16)$$

$$\pi(u) = \frac{p_{r(u)}}{n! p(u)} \prod_{s \in c(u)} \pi(s), \quad (17)$$

$$\pi(u) = \frac{1}{n! \prod_{t \subseteq u} p(t)} \prod_{i \in u} p_i. \quad (18)$$

It must be recalled that $i \in u$ means that i is a node in the tree u , that $s \in c(u)$ means that s is a subtree of the root of u and that $t \subseteq u$ means that t is any subtree of u . The last formula (18) is not inductive and is, indeed, the announced closed formula for the asymptotic probability of visit of each state.

Proof. The stationary state distribution π_p of the probabilistic path reversal algorithm over $A(L)$ is simply obtained from the invariant measure λ_p by division by $\sum_{v \in A(L)} \lambda(v)$, which will be shown to satisfy

$$\sum_{v \in A(L)} \lambda(v) = \frac{(|L| - 1)!}{\prod_{i \in L} p_i}. \quad (19)$$

with $L = \text{support}(p)$. The detailed proof, somewhat technical, is reported at the end of this article. It leads to (16) in Theorem 6 and formulae (17) and (18) then follow from Theorem 5. \square

6. Performance analysis

There are numerous indicators of performance for the path reversal algorithm, sometimes depending on the applicative context. The present study just concentrates on one

indicator, namely the average-case computational complexity, since it can be used in many applications, as will be seen in Section 6.5.

The computational cost of a single path reversal step (see Fig. 1) is estimated by the number of redirected edges when the node x_k becomes the root. It can be noted that this cost always equals the height $h(x_k, t)$ of node x_k in t , where the height of the root is zero and the height of any other node is one plus the height of its father.

Then, for every label set L , the average cost when leaving a tree t in $A(L)$ is simply

$$M(t) = \sum_{i \in L} p_i h(i, t),$$

where p_i is the probability that node i becomes the root. Finally, in the stationarity context where the steady-state probability distribution is π , the average-case computational complexity C_L can be obtained by

$$C_L = \sum_{t \in A(L)} \pi(t) M(t).$$

The Markov chain satisfies all the required conditions that give existence to this asymptotical complexity, as the limit of an average complexity after m steps, when m tends to ∞ . In particular, the connexity in the state space implies that this limit does not depend on the initial tree.

6.1. The cost distribution

The average-case complexity C_L of path reversal over L -labeled trees can also be seen as the first moment of the random variable X_L that counts the height of the reversing node, when the path reversal follows its asymptotic behavior. More formally,

$$C_L = \sum_{k \geq 0} k \mu_{k,L}, \quad (20)$$

where $\mu_{k,L} = \Pr(X_L = k)$ is the probability that a node of height k becomes the root through a single path reversal step.

This section aims to establish this discrete probability distribution of X_L , called the *cost distribution*. This distribution is finite, since it is obviously zero for heights exceeding the number of tree nodes. For $k \geq |L|$, $\mu_{k,L} = 0$.

Otherwise, the probability $\mu_{k,L}$ can be made explicit by a double enumeration over trees in $A(L)$ and over nodes in these trees. It is the total probability that the system leaves a tree t housing a node i at height k to form a tree t' where i is the root. In its turn, this event independently means that the system is in state t (which is verified with probability $\pi(t)$) and that the node chosen is i (which is verified with probability $p_i/p(L)$). These considerations lead to

$$\mu_{k,L} = \sum_{t \in A(L)} \sum_{\substack{i \in t \\ h(i,t)=k}} \frac{p_i}{p(L)} \pi(t) \quad (21)$$

or equivalently to

$$\mu_{k,L} = \sum_{i \in L} \frac{p_i}{p(L)} \sum_{\substack{t \in A(L) \\ h(i,t)=k}} \pi(t). \quad (22)$$

The useful properties of $\mu_{0,L}$ are summarized in the following lemma.

Lemma 7. *The asymptotic probability $\mu_{0,L}$ that one step of the stochastic path reversal algorithm has no effect is*

$$\mu_{0,L} = \frac{\sum_{i \in L} p_i^2}{p(L)^2}. \quad (23)$$

This probability reaches its minimum $1/|L|$ in the equiprobabilistic case, when $p_1 = \dots = p_{|L|} = 1/|L|$. In the general case,

$$\mu_{0,L} \geq \frac{1}{|L|}. \quad (24)$$

Proof. For every label i in L , the condition $h(i,t)=0$ is equivalent to $r(t)=i$ and it can be derived from Lemma 11 (or from the meaning of stationarity) that the total probability to find i at the root of the current tree-state is $p_i/p(L)$, which leads to Eq. (23). The equiprobabilistic case is a simple application and the lower bound in the general case comes from the development of $\sum_{i \in L} p_i^2$ and from the well known inequality $2p_i p_j \leq p_i^2 + p_j^2$ when $0 \leq p_i, p_j \leq 1$. \square

6.2. A recurrence formula

Now, for $k \geq 1$, there is a recurrence relation between $\mu_{k,L}$ and $\mu_{k-1,S}$ (with $S \subset L$). The key is to decompose any tree t in $A(L)$ into two smaller ones and introduce the following inductive definition of the stationarity distribution π

$$\pi(s/u) = \frac{(|S| - 1)! (|U| - 1)!}{(|S| + |U| - 1)!} \frac{p(U)}{p(S) + p(U)} \pi(s) \pi(u), \quad (25)$$

that is derived from (10) and (16), for every s in $A(S)$ and every u in $A(U)$.

The argument to decompose any tree $t \in A(L)$ is quite similar to the one used in the proof of Section 7, although even simpler. For every $k \neq 0$ and for every label i in L , $h(i,t)=k$ implies that i is not the root of t , and so determines a unique direct subtree s of t housing i as node label. This suggests a unique decomposition of t into s that is in some $A(S)$ and $t-s$ that is in $A(L-S)$, with the constraints $S \subset L$, $i \in S$ and $L-S \neq \emptyset$.

As $h(i,s)=h(i,t)-1$ the following can be derived:

$$\mu_{k,L} = \sum_{i \in L} \frac{p_i}{p(L)} \sum_{\{i\} \subset S \subset L} \sum_{\substack{s \in A(S) \\ h(i,s)=k-1}} \sum_{u \in A(L-S)} \pi(s/u) \quad (26)$$

from (22). Injecting (25) leads after some calculations to

$$\mu_{k,L} = \frac{1}{p(L)^2} \sum_{\emptyset \subsetneq S \subsetneq L} \left\{ \frac{|L|}{|S|} \right\} p(S) p(L-S) \mu_{k-1,S} \quad (27)$$

with

$$\left\{ \frac{k}{q} \right\} = \frac{(q-1)!(k-q-1)!}{(k-1)!}, \quad (28)$$

which is the announced recurrence formula.

In the equiprobabilistic case where $p_i = 1/|L|$ for all $i \in L$ and for a label set L of large size $|L| = n$, it can be shown that $\mu_{k,L} \sim n^{2k-1}(\log n)^k$.

6.3. Average-case complexity

From the knowledge of the cost distribution, a recursive definition can be derived for the average-case complexity C_L and its properties shown.

Lemma 8. *For every transition measure $p = (p_i)_{i \in L}$, the average-case computational complexity $C_L(p)$ of the stochastic path reversal over $A(L)$ is inductively defined by $C_L = 0$ if $|L| = 1$ and*

$$C_L = \frac{1}{p(L)^2} \sum_{\emptyset \subsetneq S \subsetneq L} \left\{ \frac{|L|}{|S|} \right\} p(S) p(L-S) (C_S + 1) \quad (29)$$

if $|L| \geq 2$.

Proof. In the basic case where $|L| = 1$ no computation is performed, since there is a single execution state. Now, for two or more nodes, (29) merely follows from (20) and (27). \square

6.4. A tight upper bound

This section gives a tight upper bound for the average-case complexity in the following theorem, where

$$H_n = \sum_{1 \leq k \leq n} \frac{1}{k}$$

denotes the n th harmonic number.

Theorem 9. *The average-case computational complexity $C_n(p)$ of a stochastic path reversal based on the transition distribution $p = (p_i)_{1 \leq i \leq n}$ satisfies*

$$C_n(p) \leq H_n - 1 \quad (30)$$

and the inequality in (30) is an equality in the equiprobabilistic case where $p_1 = \dots = p_n = 1/n$.

Proof. The extension

$$C_L(p) \leq H_{|L|} - 1 \quad (31)$$

of (30) to any label set L and any node measure p (it is no longer assumed that $p(L)=1$) is proven with the induction hypothesis

$$\forall S, \emptyset \subsetneq S \subsetneq L \Rightarrow C_S \leq H_{|S|} - 1 \quad (32)$$

where C_S is the average-case computational complexity of the stochastic path reversal performed over $A(S)$ for any transition measure.

As usual, the proof is performed by induction over label sets ordered by inclusion. For the basic case where $|L|=n=1$, it has been seen that $C_L=C_n=0$ and so verifies (30). Otherwise, for any label set L with two or more labels, the induction hypothesis (32) can be injected into the recursive definition (29) to obtain

$$C_L \leq \frac{1}{p(L)^2} \sum_{s=1}^{|L|-1} \left\{ \begin{matrix} |L| \\ s \end{matrix} \right\} H_s \sum_{\substack{S \subset L \\ |S|=s}} p(S) p(L-S). \quad (33)$$

For any distinct labels i and j in L , the term $p_i p_j$ appears in

$$\sum_{\substack{S \subset L \\ |S|=s}} p(S) p(L-S)$$

for each subset S with $|S|=s$, $i \in S$ and $j \in L-S$, or $j \in S$ and $i \in L-S$. There are exactly

$$\binom{|L|-2}{s-1} = \frac{(|L|-2)!}{(s-1)!(|L|-s-1)!}$$

such sets, so that

$$\sum_{\substack{S \subset L \\ |S|=s}} p(S) p(L-S) = \binom{|L|-2}{s-1} \left(p(L)^2 - \sum_{i \in L} p_i^2 \right) \quad (34)$$

and

$$C_L \leq \frac{1}{|L|-1} \left(1 - \frac{\sum_{i \in L} p_i^2}{p(L)^2} \right) \sum_{s=1}^{|L|-1} H_s. \quad (35)$$

Now, the upper bound (24) for $\mu_{0,L} = \sum_{i \in L} p_i^2 / p(L)^2$ and the classical formula

$$\sum_{s=0}^{n-1} H_s = n(H_n - 1)$$

for harmonic numbers give $C_L(p) \leq H_{|L|} - 1$ and end the proof. \square

6.5. Back to the applications

The average-case complexity bounded in Section 6.4 has a meaningful interpretation in each application of path reversal described in the introduction. This section gives this interpretation for the context of distributed mutual exclusion.

The Naimi–Tréhel algorithm [10] maintains a dynamic logical rooted tree structure to perform mutual exclusion between n network nodes (see Fig. 1). The root x_1 of the tree is the node that holds the exclusive privilege to use the critical resource. Any other node x_k that needs this resource sends a requesting message to its parent in the tree. Any intermediate node x_{k-1}, \dots, x_2 on the path to the root forwards the requesting message to its own parent and makes its new parent point to the requester x_k . When the request reaches the root x_1 , this node finishes using the resource, sends a *privilege* message to x_k and sets its new parent as x_k . The whole result is a new tree where x_k is the root and holds the privilege.

Under the non-concurrent hypothesis that no other request is emitted during the complete transit of messages, the algorithm is exactly a path reversal over the logical tree structure. A good measure for its complexity in time is the average number of messages sent by request. Let M_L be its asymptotic value when L is the set of node labels.

The number of messages needed for one request is zero if the requirer is already the root, and one plus the number of moved edges otherwise, due to the additional transfer of privilege. With the notations of Section 6,

$$M_L = \sum_{k \geq 1} (k+1) \mu_{k,L} \quad (36)$$

and M_L is simply related to C_L by

$$M_L = C_L + 1 - \mu_{0,L}. \quad (37)$$

Now, Theorem 9 and its proof have two direct consequences for the asymptotic average number of messages per request.

The first, due to (24) and (30), is that, whatever the probabilities of transition are, the asymptotic average number of messages M_n of Naimi–Tréhel algorithm between n nodes verifies

$$M_n \leq H_{n-1}, \quad (38)$$

which was a conjecture only proved up to $n=3$ in [9].

The second consequence is a more accurate upper bound depending on the transition distribution p , and more precisely on its variance $v(p)$ defined by

$$v(p) = \frac{1}{|L|} \sum_{i \in L} \left(\frac{p_i}{p(L)} \right)^2 - \left(\frac{1}{|L|} \sum_{i \in L} \frac{p_i}{p(L)} \right)^2, \quad (39)$$

where $L = \text{support}(p)$. The variance is an estimator of the difference between values of the transition probabilities $p_i/p(L)$. In eq. (35), that has been proved for $|L| \geq 2$,

can be used to obtain

$$M_L \leq H_{|L|-1} \left(1 - v(p) \frac{|L|^2}{|L|-1} \right), \quad (40)$$

which is the second conjecture of [9] that was empirically deducted by simulation. This upper bound suggests that the average number of messages per request could be much lower than $H_{|L|-1}$ when nodes require the shared resource at very different frequencies. This bound confirms the intuitive idea that the most frequent requesters tend to stay near the root of the logical tree and then request at low cost.

7. Detailed proof

This section provides a structured and detailed proof of Theorem 6. It starts with a deterministic decomposition of labeled rooted trees and continues with an intermediate summation lemma, which is then proven by induction. The proof of (19) simply appears as a corollary of the summation lemma.

7.1. A decomposition of labeled rooted trees

Let L be a label set and let $i \in L$ be a node label. In the set L , it is always possible to associate to i a uniquely defined companion i' with $i' \neq i$. For instance, fix a circular permutation σ over L and choose $i' = \sigma(i)$. In all that follows, i' denotes this node label which is uniquely associated with i .

Let $A(L, i)$ denote the set

$$A(L, i) = \{t \in A(L). r(t) = i\}$$

of labeled rooted trees with the root labeled by i .

Any tree t in $A(L, i)$ owns a unique direct subtree s_t with the companion i' of the root label $r(t) = i$ of t among its node labels. This remark suggests the following decomposition lemma, where \sum denotes disjoint set union and \leftrightarrow a bijection between sets.

Lemma 10. *For every label set L ,*

$$A(L) \leftrightarrow \sum_{i \in L} A(L, i) \quad (41)$$

and

$$A(L, i) \leftrightarrow \sum_{\{i'\} \subset S \subset L - \{i\}} A(S) \times A(L - S, i). \quad (42)$$

Proof. Every tree t in $A(L)$ is in the unique set $A(L, i)$ such that $i = r(t)$ and can therefore be uniquely decomposed into (and recomposed from) the two labeled rooted trees s_t and $t - s_t$.

Now, s_t has a set S of node labels including i' but excluding i , which is the root label of $t - s_t$. The node labels in $t - s_t$ are obviously the labels in L , but not in S . The recomposition is unique, since it just adds an S labeled rooted tree to the set of direct subtrees of some $L - S$ labeled and i rooted tree. \square

It can be noted in passing that (42) gives a combinatorial explanation of the elegant combinatorial distribution

$$n^{n-2} = \sum_{k=1}^{n-1} \binom{n-2}{k-1} k^{k-1} (n-k)^{n-k-2} \quad (43)$$

of the number n^{n-2} of (unrooted) labeled trees with n nodes counted by Cayley [2].

7.2. A summation lemma

Lemma 11. *For every label set L and for every label i in L , the formula*

$$\sum_{t \in A(L,i)} \lambda(t) = \frac{(|L| - 1)!}{p(L) \prod_{j \in L, j \neq i} p_j} \quad (44)$$

is a closed form for the total weight of measure λ over trees with root i and node labels in L .

With this lemma it is easy to prove the following corollary, which in turn constitutes a proof of (19) for Theorem 6.

Corollary 12. *For every label set L , the formula*

$$\sum_{t \in A(L)} \lambda(t) = \frac{(|L| - 1)!}{\prod_{j \in L} p_j} \quad (45)$$

is a closed form for the total weight of measure λ over trees with node labels in L .

Proof. The corollary immediately follows from Lemma 11 and (41). \square

7.3. Inductive proof of Lemma 11

The proof of Lemma 11 proceeds by induction on label sets ordered by inclusion.

Proof. For the basic case where $L = \{i\}$ for some $i \in \mathbb{N} - \{0\}$, $A(\{i\})$ is reduced to the unique rooted tree with a single node labeled by i and (44) immediately follows from (9).

Now, for every label set L with two or more labels, (42) in Lemma 10 and the relation (10) are used to decompose the total weight $\sum_{t \in A(L,i)} \lambda(t)$ of λ over $A(L,i)$ trees in

$$\sum_{\{i'\} \subset S \subset L - \{i\}} \sum_{s \in A(S)} \sum_{u \in A(L-S,i)} \frac{p(L-S)}{p(L)} \lambda(s) \lambda(u). \quad (46)$$

Then, factorization leads to

$$\frac{1}{p(L)} \sum_{\{i'\} \subset S \subset L - \{i\}} \left(\sum_{s \in A(S)} \lambda(s) \right) p(L - S) \left(\sum_{u \in A(L - S, i)} \lambda(u) \right), \quad (47)$$

where the first parenthesized sum is the left-hand side of (45) in Corollary 12 and the second parenthesized sum is the left-hand side of (44) in Lemma 11. By induction hypothesis, these sums, applied to strict subsets of L , can be replaced by their closed form to provide

$$\frac{1}{p(L)} \sum_{\{i'\} \subset S \subset L - \{i\}} \frac{(|S| - 1)! (|L - S| - 1)!}{\prod_{j \in S} p_j \prod_{j \in L - S, j \neq i} p_j} \quad (48)$$

and the expected result (44) follows after some calculations. \square

8. Conclusion

In this paper, a performance analysis has been carried out for time complexity of path reversal, a common operation on tree structures. This analysis is neither based on simulation nor on approximation, but on exact computations. These computations have been successfully treated in a general way, since they hold for any tree size and for any stochastic evolution.

After recalling why the algorithm has a stationary asymptotic behavior, new formulae are given to compute the stationary state distribution that fully determines this behavior. These inductive or direct formulae on the transition probabilities are all computable in time proportional to the tree size. The other new results are an inductive definition for the cost distribution by node height and a tight upper bound for the average time complexity.

From an applicative point of view, these results show their accuracy by solving two conjectures concerning the average number of messages in a distributed mutual exclusion algorithm based on path reversal.

This paper also shows that a proper induction over non planar trees can give access to accurate quantitative performance results for path reversal or similar algorithms. For this question as well as for others, an additional structure like ordering in the node children is not required, formal power series could surely help but are not really needed and all the computations can be kept readable without skipping fundamental steps. Moreover, it has been shown that the indeterminism hidden behind the choice of one arbitrary subtree in a non planar tree does not prevent the derivation of inductive proofs and sometimes even simplifies them. The aims of remaining close to the original problem and providing reader friendly notations have been achieved.

The same non classical approach and notations could also be applied to other algorithms over tree data structures, when they are enriched by randomized input.

Acknowledgements

The author would like to thank Professor Michel Tréhel who introduced him to the conjecture over the average number of messages in his mutual exclusion algorithm and Dr Pierre Gradiat who refined this conjecture by the use of accurate numerical simulations.

References

- [1] J. Bernabeu-Auban, M. Ahamad, Applying a Path-Compression Technique to Obtain an Efficient Distributed Mutual Exclusion Algorithm, *Lecture Notes in Computer Science*, vol. 392, Springer, Berlin, 1989, pp. 33–44.
- [2] A. Cayley, A theorem on trees, *Quart. J. Math. Oxford Ser.* 23 (1889) 376–378.
- [3] D. Ginat, D.D. Sleator, R.E. Tarjan, A tight amortized bound for path reversal, *Inform. Process. Lett.* 31 (1989) 3–5.
- [4] D.E. Knuth, *The Art of Computer Programming I: Fundamental Algorithms*, Addison-Wesley, Reading, MA, 1968.
- [5] C. Lavault, Analysis of an efficient distributed algorithm for mutual exclusion (average-case analysis of path reversal), *Lecture Notes in Computer Science*, vol. 634, Springer, Berlin, 1992, pp. 133–144.
- [6] M. Naimi, M. Tréhel, A. Arnold, A $\log(N)$ distributed mutual exclusion algorithm based on path reversal, *J. Parallel Distributed Comput.* 34 (1996) 1–13.
- [7] J.R. Norris, *Markov Chains*, Cambridge University Press, Cambridge, 1997.
- [8] R.E. Tarjan, J. van Leeuwen, Worst-case analysis of set union algorithms, *J. ACM* 31 (2) (1984) 245–281.
- [9] M. Tréhel, P. Gradiat, A. Giorgetti, Performances d’un algorithme distribué d’exclusion mutuelle en cas de non-équiprobabilité des requêtes des processus, *Réseaux et Systèmes Répartis Calculateurs Parallèles*, numéro spécial “Evaluation quantitative des performances des réseaux et systèmes”, vol. 13(6) Hermès Science, Paris, 2001, pp. 557–573.
- [10] M. Tréhel, M. Naimi, Un algorithme distribué d’exclusion mutuelle en $\text{Log}(n)$, *Tech. Sci. Inform.* 6 (1987) 141–150.
- [11] J. van Leeuwen, T. van der Weide, Alternative path compression techniques, Technical Report RUU-CS-77-3, Department of Computer Science, University of Utrecht, Utrecht, The Netherlands, 1977.